

논문 2012-50-3-14

모니터 화면 내 활성화된 동영상 재생기 영역 검출 기법

(Detection Algorithm of an Active Video Player Region in the Monitor Screen)

김 학 구*, 송 병 철**

(Hak Gu Kim and Byung Cheol Song)

요 약

본 논문은 모니터나 스마트TV 등에서 재생 중인 동영상 재생기의 창 영역을 정확하게 검출하는 기법을 제안한다. 단순히 시간적 차이 유무로 판단하거나 운영체제로부터 정보를 가져오는 기존 기법들과 달리, 제안하는 동영상 재생기 검출 기법은 영상 신호의 시공간적 특성 및 꼭지점 검출 필터를 이용하여 매우 정확하게 재생기 영역을 검출할 수 있다. 먼저, 영상 신호의 시간적 특성을 이용하여 재생기 창의 초기 영역을 설정한 뒤, 그 초기 영역을 기반으로 영상의 공간적 특성인 에지 정보와 꼭지점 검출 필터를 이용하여 재생기 창 영역의 네 꼭지점들의 위치를 계산한다. 모의실험을 통해 제안하는 기법이 빠른 연산 속도와 정확한 검출 성능을 보임을 확인하였다.

Abstract

This paper presents a detection algorithm that accurately finds the active area of a video player on monitors or smart TVs. Unlike the previous approaches, temporal difference-based detection algorithms or hooking programs, the proposed detection algorithm can locate the active video player by using the spatial and temporal correlation and a corner detection filter. First, an initial location of the video player is found using conventional temporal difference-based detection. Then, starting from the initial location, the four corners of the active video player are detected by the spatial edge information and the corner detection filter. The experimental results show that proposed algorithm provides fast detection speed and high accuracy.

Keywords : temporal difference, corner detection, active video player, edge.

I. 서 론

최근 인터넷과 고해상도 카메라가 보편화됨에 따라 사용자 제작 영상 (UCC), 영화 등 다양한 영상 콘텐츠들이 급속히 증가하고 있다. 또한, 웹 하드와 클라우드 등을 통해 콘텐츠 공유가 활발해지면서 TV보다 컴퓨터에서 영상 콘텐츠를 접하는 사람들이 점점 늘어나고 있다. 그러나 모니터를 통해 동영상 콘텐츠를 접할 때

는 몇 가지 불편한 점들이 있다. 첫째, 동영상의 화질 열화 문제다. 동영상 콘텐츠들은 방대한 데이터 용량 때문에 일반적으로 MPEG-2, AVC/H.264 등으로 압축되어 저장되거나 전송된다^[1~2]. 이렇게 압축된 동영상은 데이터 손실로 인해 모니터 성능에 관계없이 화질의 한계가 있다. 특히, 사용자가 직접 찍어 올린 웹 동영상 콘텐츠들은 화질 열화가 심한 편이다. 둘째는 컴퓨터 고유의 사용 환경에 따른 불편함이다. 사람들이 컴퓨터로 단순히 동영상만 보는 경우는 드물다. 예를 들어 문서 작업을 하다가 영화를 보기도 하고 인터넷 쇼핑물창을 띄워놓은 채 또 다른 창에서는 유튜브 영상을 보기도 한다. 이처럼 모니터 화면에 많은 창들이 열려있거나 배경 화면이 복잡한 경우, 사용자는 동영상 화면에만 집중하기 어렵다.

* 학생회원, ** 평생회원, 인하대학교 전자공학부 (School of Electronic Engineering, Inha University)

※ “이 논문은 2012년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구이며 (2012R1A1B3000446), LG전자의 지원을 받아 수행된 결과임.”

접수일자:2012년8월15일, 수정완료일:2013년2월18일

최근 화면 내에서 동영상이 재생되고 있는 영역을 찾아 이 부분만 따로 영상 처리를 해주는 기술이 적용된 모니터들이 출시된 바 있다. 예를 들면, 동영상 창 이외 영역은 어둡게 하여 사용자의 몰입도를 높여주거나 더 나아가 영상이 재생되는 영역의 색감이나 선명도를 개선하여 시각적인 효과를 높여주는 기능 등이 있다.

상기 기술의 핵심은 영상의 재생 영역을 정확히 검출하는 것이다. 지금까지는 인접 프레임 간의 휘도 차이를 관찰하여 동영상 재생 영역을 검출하거나 운영체제로부터 실행프로그램 위치 정보를 빼내오는 프로그램을 이용하는 방법들이 사용되었다^[3-4].

전자의 방법 [3]은 인접한 두 프레임 간의 휘도차이가 40이상인 영역을 동영상이 재생되고 있는 영역으로 가정한다. 그리고 60프레임 동안 충분히 관찰하여 20회 이상 40이상의 휘도차이가 발생하는 영역을 최종 동영상영역으로 판단한다. 이 기법은 매우 간단하고 연산량이 적은 장점이 있다. 하지만 마우스 움직임이나 팝업창 등 여러 잡음에 민감하고, 문턱치 값에 매우 의존적인 방법으로 다양한 상황에서 창의 꼭지점 위치를 정확하게 검출할 수 없다. 방법 [4]는 많은 컴퓨터가 윈도우 운영체제라는 점을 이용한 방법이다. 윈도우 운영체제는 프로그램 창마다 해당 창에 대한 정보를 담고 있는 윈도우 메시지 핸들러라는 것이 존재한다. [4]의 방법은 동영상 재생기가 실행되면 재생기에 대한 핸들러에 접근하여 재생기 위치에 대한 정보를 빼내오는 것이다. 그러므로 가장 정확한 반면 여러 제약조건이 따른다. 우선, 운영체제로부터 재생기 정보를 가져오는 프로그램과 좌표 검출 프로그램을 동영상 플레이어보다 먼저 실행시켜야 하는 점이다. 그리고 윈도우라는 특정 운영체제에서만 적용된다는 한계가 있다. 마지막으로 인터넷에서 바로 볼 수 있는 콘텐츠에는 적용되지 않는다.

이러한 단점들을 극복하기 위해 본 논문은 영상 신호의 시공간 특성을 모두 고려한 동영상 재생기 검출 알고리즘을 제안한다. 제안하는 방법은 크게 가상 영역 생성 단계와 꼭지점 검출 단계로 이루어져 있다. 먼저, 가상 영역 생성 단계에서는 시간에 따른 밝기 변화를 관찰하여 재생기의 초기 영역 즉, 가상 영역의 위치와 크기를 구한다. 그런 다음, 꼭지점 검출 단계에서 가상 영역을 바깥으로 확장해 가면서 재생기의 정확한 네 꼭지점들을 탐색한다. 네 개의 꼭지점들이 모두 찾아지면 이 꼭지점들로 이루어지는 사각형 영역을 최종 재생기 영역으로 결정한다.

본 논문의 구성은 다음과 같다. 다음 II절에서는 제

안 기법을 구체적으로 설명한다. 그리고 III절에는 제안하는 방법의 실험 환경과 결과 영상들을 담았으며, 마지막 IV절은 본 논문의 결론으로 구성되어 있다.

II. 제안 기법

본 논문은 화소 값들의 시공간적 특성 정보와 꼭지점 검출 필터를 이용하여 동영상 재생기 영역을 추출하는 방법을 제안한다. 제안 기법은 다음의 두 가지 특징을 갖는다. 첫째, 가상 영역 생성 단계에서 시간적 차이 정보와 시간적 일관성을 이용해 재생기의 초기 위치를 효과적으로 찾는다. 둘째, 상기 초기 위치로부터 탐색을 시작하여, 꼭지점 검출 필터와 영상의 에지 정보를 이용해 재생기의 네 꼭지점들의 위치를 정확히 계산한다. 결과적으로 제안 기법은 재생기 창의 이동이나 크기 변경에도 강인하게 그 위치를 정확하게 검출할 수 있다.

그림 1은 제안 기법의 순서도이며, 각 단계 별 세부 내용은 아래와 같다.

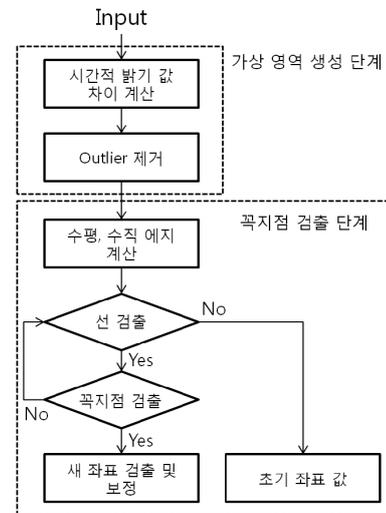


그림 1. 제안하는 동영상 재생기 검출 방법
Fig. 1. Block diagram of the proposed video player detection algorithm.

1. 가상 영역 생성

가. 시간적 밝기 변화 계산

제안 기법은 먼저 영상 신호의 시간적 밝기 변화 특성을 이용하여 재생기의 초기 위치를 계산한다. 시간에 따른 밝기 변화를 아래 식 (1)과 같이 (x, y)위치에서의 인접한 프레임 간의 밝기 차이 $D(x,y)$ 를 구하여 가상

영역을 검출한다.

$$D(x,y) = |I_t(x,y) - I_{t-1}(x,y)| \quad (1)$$

여기서 $I_t(x,y)$ 와 $I_{t-1}(x,y)$ 은 각각 t 번째 프레임과 $t-1$ 번째 프레임 내 (x, y) 위치에서의 밝기를 의미한다. 그런 다음, 식 (2)와 같이 시간 축 변화를 나타내는 이진 맵을 생성한다.

$$T_d(x,y) = \begin{cases} 255, & D(x,y) > 0 \\ 0, & D(x,y) = 0 \end{cases} \quad (2)$$

그림 2는 생성된 $T_d(x,y)$ 영역의 한 예를 보인다. 그림 2 (b)를 보면 사용자의 마우스 조작 등에 의해 재생 영역 밖에서도 시간적 차이가 발생하는 것을 알 수 있다. 우리는 이런 영역들을 잡음으로 간주하고 다음 단계에서 이를 제거한다.

나. 잡음 제거

그림 2의 예처럼, 모니터라는 환경 때문에 동영상 재생기 영역 밖에서도 얼마든지 변화가 있을 수 있다. 예를 들면, 마우스를 움직일 때나 아이콘을 클릭했을 때, 또는 팝업창이 떴을 때 등의 상황들이다. 따라서 동영상 재생기 영역을 제외한 나머지 변화가 있는 영역을 잡음으로 규정하고 제거해야 한다.

본 논문에서는 닫힘 연산과 시간적 일관성을 이용하여 다양한 잡음들을 제거한다. 닫힘 연산은 팽창과 침식을 순서대로 수행하는 형태학적 방법을 말한다. 먼저,

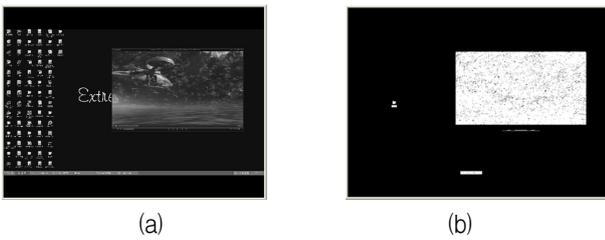


그림 2. (a) 원본 영상, (b) 시간적 밝기 차이 영역
Fig. 2. (a) Original image, (b) temporal difference image.

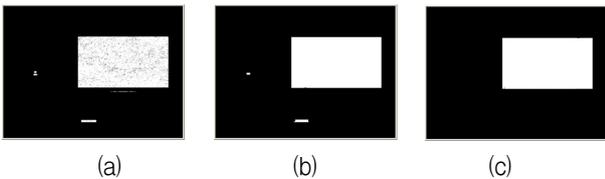


그림 3. (a) 입력 시간적 차이 영상, (b) 닫힘 연산 수행 결과, (c) 잡음들이 모두 제거된 후 가상 영역
Fig. 3. (a) Input temporal difference image, (b) the result after closing operation, (c) virtual box after removing all outliers.

닫힘 연산을 수행하여 짧게 끊어진 영역과 작은 구멍들을 메우고, 동 떨어져 있는 작은 영역들의 크기를 축소하거나 제거한다^[5]. 즉, 닫힘 연산을 통해 그림 2에서 보이는 작은 구멍들은 메워지고 따로 떨어져있는 작은 크기의 잡음을 제거될 수 있다. 그림 3은 닫힘 연산을 수행하고 난 뒤의 시간적 차이 영상 $T_d'(\cdot)$ 의 예이다.

닫힘 연산만으로 제거될 수 없는 크기의 나머지 잡음들을 제거하기 위해 소위 시간적 변화의 일관성을 관찰한다. 잡음들은 대개 지속적으로 변하는 것이 아니라 일시적인 변화를 가지므로 같은 위치에서 변화가 지속적으로 발생하는지를 관찰하면 이를 검출할 수 있다. 먼저, 식 (3)을 통해 인접한 프레임 간 시간적 차이 영역의 변화 여부에 따라 가중치 $cnt(x,y)$ 를 부과한다.

$$cnt_t(x,y) = \begin{cases} +1, & T_{d,t-1}'(x,y) = T_{d,t}'(x,y) \\ -1, & T_{d,t-1}'(x,y) \neq T_{d,t}'(x,y) \end{cases} \quad (3)$$

$t-1$ 번째 프레임 내 (x, y) 위치에서 시간적 차이와 t 번째 프레임 내 (x, y) 위치에서 시간적 차이가 같으면 1, 다르면 -1을 취한다. 즉, $cnt_t(x,y)$ 는 인접한 프레임 간의 시간적 차이 영역이 동일한지를 판별하는 변수이다. 그리고 식 (4)와 같이 소정의 시간동안 가중치들을 합산하여 소위 시간적 일관성 $T_c(\cdot)$ 를 계산한다.

$$T_c(x,y) = \sum_{i=0}^4 cnt_{t-i}(x,y) \quad (4)$$

본 논문에서는 5프레임 동안의 변화를 관찰한다.

만약 변화가 꾸준히 발생한다면 시간적 일관성 $T_c(\cdot)$ 값은 커지게 된다. 최종적으로 식 (5)를 통해 시간적 일관성이 큰 화소들만으로 이루어진 새로운 시간적 차이 영역 $T_d''(\cdot)$ 을 생성한다.

$$T_d''(x,y) = \begin{cases} T_d'(x,y), & T_c(x,y) > 3 \\ 0, & otherwise \end{cases} \quad (5)$$

즉, $T_d''(\cdot)$ 은 시간적 일관성 $T_c(\cdot)$ 값이 작은 영역들을 제거하고 남은 최종 시간적 차이 영역이다. 그림 3 (c)는 잡음들이 모두 제거된 영상의 예이다.

다. 후처리 및 초기 좌표 설정

한편, 배경 변화는 거의 없이 물체들만 움직임이 있는 영상들이 존재하는데 주로 카메라를 고정시켜놓고 실내에서 촬영한 장면들이다. 이런 경우, 특정 프레임에서 해당 $T_d''(\cdot)$ 영상으로부터 충분한 크기의 가상 영

역이 생성되지 않을 수 있다. 이런 문제를 해결하기 위해 후처리로써 이전 프레임의 $T_{d,t-1}(\cdot)$ 를 현재 프레임의 $T_{d,t}(\cdot)$ 와 병합한다.

가상 영역이 최종적으로 결정되면 그 가상 영역의 네 꼭지점들을 동영상 재생기의 초기 좌표로 설정한다. 실제로 가상 영역은 화면 내에서 지속적인 변화가 발생하는 영역이므로 실제 동영상 영역과도 매우 흡사하다. 그래서 꼭지점 검출 단계에서 실제 동영상 영역의 좌표가 구해지지 않는 경우, 이 초기 좌표를 최종 좌표로 대체한다.

2. 꼭지점 검출

최종적인 동영상 재생기 영역 검출은 선 검출과 꼭지점 검출로 이루어진다.

가. 선 검출

먼저 소벨 연산자를 이용해 수평, 수직 방향의 에지를 검출하고 이진화하여 각 에지 맵 $S_x(x,y), S_y(x,y)$ 를 만든다^[5]. 그리고 식 (6), (7)과 같이 에지 맵 내에서 가상 영역을 키워나가며 선 에지를 찾는다.

$$t_{hline} = \sum_{i=x_1}^{x_2} S_x(i,y) \quad (6)$$

$$t_{vline} = \sum_{j=y_1}^{y_2} S_y(x,j) \quad (7)$$

그림 4는 선 검출 방법을 개념적으로 나타낸 그림이다. 가상 영역의 상하좌우 네 변의 위치를 y_1, y_2, x_1, x_2 라고 할 때, 식 (6), (7)을 통해 계산된 t_{hline} 과 t_{vline} 값이 τ_1, τ_2 이상일 때 해당 방향의 선 에지로 판단한다.

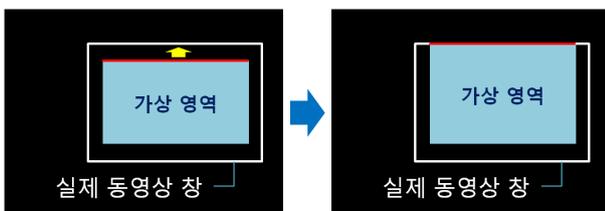


그림 4. 선 검출 방법
Fig. 4. The line edge detection.

나. 꼭지점 검출

지금까지 해리스 코너, SIFT 등 꼭지점 및 특징점을

검출하는 기법들은 많이 개발되었다^[6~7]. 하지만 이러한 꼭지점 검출 기법들은 모니터 화면에 적용하면 아이콘과 배경 화면 등에서 불필요한 꼭지점들이 너무 많이 검출될 뿐만 아니라 연산량이 방대하여 실시간 동작이 거의 불가능하다.

제안하는 꼭지점 검출 방법은 그림 5와 같다. 모서리

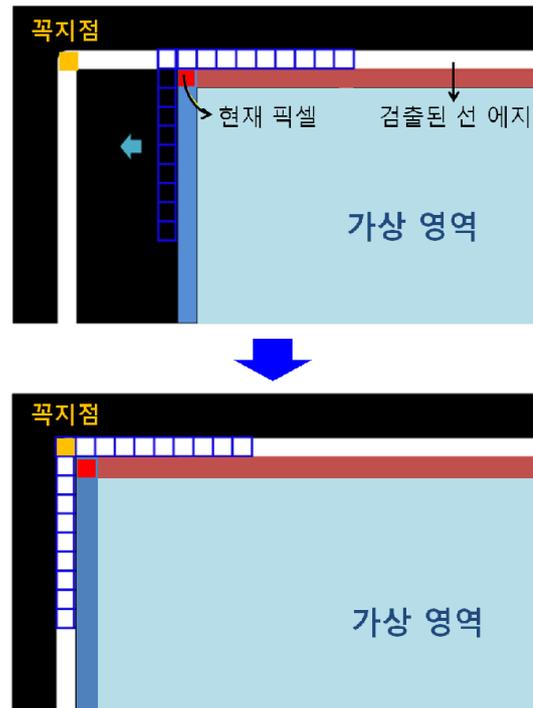


그림 5. 꼭지점 검출 방법
Fig. 5. The corner detection.

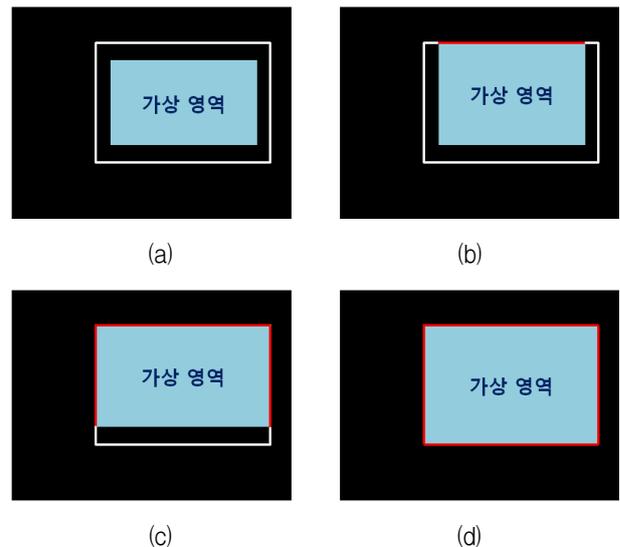


그림 6. (a) 초기 상태, (b) 윗부분 선 검출, (c) 윗부분 꼭지점 검출, (d) 아랫부분 선과 꼭지점 검출
Fig. 6. (a) Initial station, (b) upper line edge detection, (c) upper corner detection, (d) lower line and corner detection.

의 형태에 따라 그림 5에서 과란색으로 표시한 것과 같은 격자형 꼭지점 검출 필터를 적용하여 수평, 수직 에지가 모두 존재하는지 판단한다.

네 개의 꼭지점 검출 필터는 아래 식 (8)-(11)과 같다.

$$c_{tl} = \sum_{i=-1}^8 S_x(x+i, y-1) + \sum_{j=-1}^8 S_y(x-1, y+j) \quad (8)$$

$$c_{tr} = \sum_{i=-1}^8 S_x(x-i, y-1) + \sum_{j=-1}^8 S_y(x+1, y+j) \quad (9)$$

$$c_{bl} = \sum_{i=-1}^8 S_x(x+i, y+1) + \sum_{j=-1}^8 S_y(x-1, y-j) \quad (10)$$

$$c_{br} = \sum_{i=-1}^8 S_x(x-i, y+1) + \sum_{j=-1}^8 S_y(x+1, y-j) \quad (11)$$

현재 화소의 위치가 (x, y)일 때 왼쪽 또는 오른쪽으로 이동하며 꼭지점을 검출하게 된다. 여기서, c_{tl} 은 좌상단 꼭지점, c_{tr} 은 우상단 꼭지점, c_{bl} 은 좌하단 꼭지점, c_{br} 은 우하단 꼭지점을 검출하기 위한 필터들의 결과치들이다.

본 논문에서 제안하는 꼭지점 검출의 순서는 그림 6과 같다. 먼저, 그림 6 (b)처럼 재생기 영역의 윗변이 검출되면 가상 영역을 좌, 우로 확장하며 왼쪽에는 식 (8), 오른쪽에는 식 (9)을 적용하여 c_{tl} , c_{tr} 값을 계산한다. c_{tl} , c_{tr} 이 τ_3 보다 큰 두 지점을 좌상단과 우상단의 꼭지점으로 결정한다 (그림 6 (c)). 그 다음에는 재생기의 아랫변을 검출하고 다시 가상 영역을 좌, 우로 확장하며 각각 식 (10), (11)을 적용하여 그 값이 τ_3 보다 큰 좌표를 좌하단과 우하단 꼭지점으로 결정한다 (그림 6 (d)).

다. 좌표 보정

재생기 창 영역 검출 기법은 무엇보다도 높은 정확도가 요구된다. 따라서 제안하는 기법은 좌표 보정을 위해 검출된 재생기 영역과 가상 영역의 크기와 위치가 매우 유사하다는 특성을 이용한다. 즉, 검출된 영역과 가상 영역의 넓이 및 중심점 위치를 비교한다. 만약 재생기의 초기 영역인 가상 영역과 최종 검출된 영역의 넓이와 위치가 서로 상이하다면, 꼭지점 검출 단계에서 재생 영역의 꼭지점이 잘못 검출된 것이므로 초기 가상 영역의 꼭지점 좌표로 대체한다.

라. 연산량 최적화

재생기의 네 꼭지점 좌표는 매 프레임 가상박스 단계와 선 검출 그리고 꼭지점 검출 단계를 거쳐 계산된다. 하지만 재생기의 위치나 크기가 변하지 않는 경우에 동일한 꼭지점 좌표를 매 프레임 새로 계산할 필요는 없다. 따라서 이전 프레임과 현재 프레임의 가상 영역을 비교하여 그 크기와 위치가 동일하다면 재생기 창 역시 변화가 없는 것으로 간주한다. 그래서 선 검출과 꼭지점 검출 단계를 생략하고 이전 프레임에서 검출된 꼭지점 좌표들로 대체하여 연산량을 최적화한다.

III. 실험

제안 기법의 성능 평가를 위해 다양한 화면, 창을 이동시키거나 크기를 변경하는 경우, 중간에 광고창이 뜨는 경우 등 여러 시나리오에 대해 실험을 진행하였다.

그림 7은 본 논문에서 사용된 다양한 실험 영상들이다. 그림 7 (a)는 배경 화면에 수평, 수직 에지가 많이 있는 경우, 그림 7 (b)는 재생기의 크기가 확대되는 경우, 그림 7 (c)는 재생기의 위치를 이동시키는 경우, 그리고 그림 7 (d)는 인터넷에 올려진 동영상을 보는 경우에 대한 영상들이다. 표 1은 각 실험 영상들의 특징

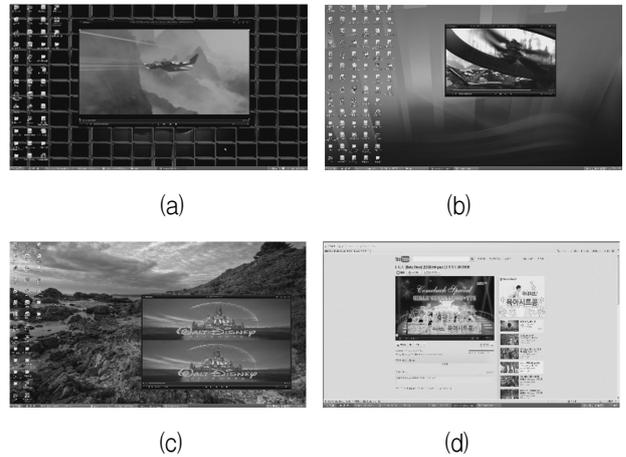


그림 7. 실험에 사용된 다양한 동영상
Fig. 7. A variety of videos for experiments.

표 1. 실험 동영상들의 특징
Table 1. Characteristics of the test videos.

	(a)	(b)	(c)	(d)
프레임 수	30	50	90	80
프레임 크기	1920x1080 @ 30 Hz			
특징	에지 배경화면	창 크기 확대	창 위치 이동	인터넷 환경

표 2. 실험에 사용한 변수들
Table 2. Parameters in experiment.

Parameters	
τ_1	$(x_2 - x_1) \times 0.7$
τ_2	$(y_2 - y_1) \times 0.7$
τ_3	14



그림 8. 그림 7 (b) 동영상에서의 재생기 검출 결과
Fig. 8. Player detection result for Fig. 7 (b).



그림 9. 그림 7 (d) 동영상에서의 재생기 검출 결과
Fig. 9. Player detection result for Fig. 7 (d).

을 기술하고 있다. 모니터 녹화 프로그램으로 HD 화면 전체를 녹화하여 실험 동영상들을 획득하였다. 동영상 재생기로는 KMPlayer가 사용되었다. 표 2는 제안 기법의 변수 값들을 정리하고 있다.

그림 8과 9는 각각 그림 7 (b)과 그림 7 (d)의 동영상에서의 재생기 검출 결과를 보인다. 그림에서 하얀색 점들이 바로 검출된 재생기의 네 꼭지점들이다. 그림 8과 9의 실험 결과를 통해 제안하는 방법이 거의 정확하게 동영상 영역의 꼭지점 좌표를 검출해내는 것을 확인할 수 있다. 유사하게 그림 7 (a)와 그림 7 (c)에 대해서도 제안하는 방법이 동영상 영역을 정확히 검출하는 것을 마찬가지로 확인하였다.

그림 10은 II.2.라 절에서 제안한 연산량 최적화 전과 후의 프레임 별 연산 속도를 비교하는 그래프이다. 최적화될 경우 재생기의 크기나 위치가 변하는 순간에만

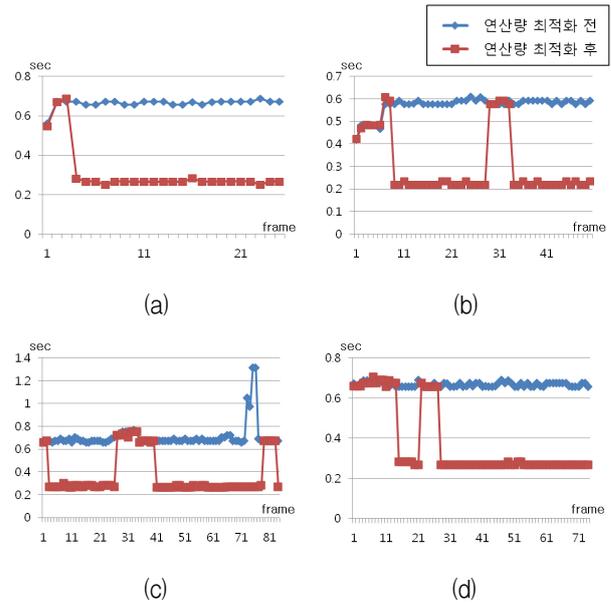


그림 10. 연산량 최적화 효과
Fig. 10. The effect of computational optimization.

표 3. 영상 별 연산 속도의 평균과 분산
Table 3. Mean and variance of processing speed per frame.

		평균 연산 속도 (s)	분산
(a)	최적화 전	0.664 s	0.022
	최적화 후	0.310 s	0.122
(b)	최적화 전	0.572 s	0.039
	최적화 후	0.304 s	0.141
(c)	최적화 전	0.705 s	0.110
	최적화 후	0.376 s	0.184
(d)	최적화 전	0.665 s	0.010
	최적화 후	0.376 s	0.178

연산 속도가 늘어나고 평균적으로는 연산량이 크게 감소한다. 표 3은 프레임 별 연산 속도의 평균과 분산을 계산한 결과를 보인다. 최적화 전에는 가상 영역 생성부터 최종 꼭지점 검출까지 프레임 당 약 0.6-0.7초가 소요됨을 알 수 있다. 그러나 연산량 최적화 후 평균 연산 시간이 0.2-0.3초로 크게 줄어들음을 확인할 수 있다.

마지막으로 본 논문은 기존 기법 [3]과 제안하는 기법의 꼭지점 검출 정확도를 비교하여 제안 기법의 성능을 확인하였다. 본 논문은 10여개의 동영상 콘텐츠들을 플레이한 후 각 알고리즘에 의해 검출된 네 개의 꼭지점 좌표와 실제 네 꼭지점 좌표 간의 유클리디안 거리의 합을 계산하여 검출 오차를 측정하였다. 그림 11은 방송, 영화, 유튜브 등 총 10개의 다양한 영상에 대한 꼭지점 검출 정확도를 보여주는 그래프이다. 제안하는

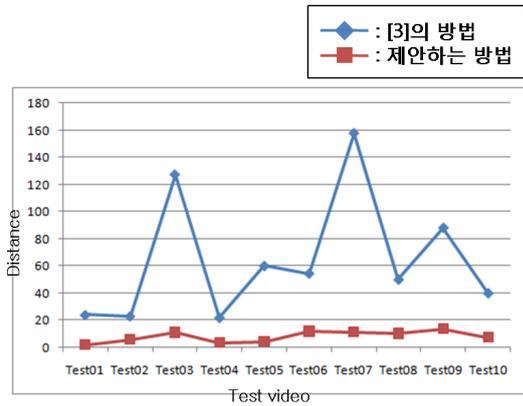


그림 11. 제안 기법과 기존 기법의 정확성 비교
 Fig. 11. Comparison between proposed and previous method [3].

기법이 기존 기법 [3]보다 다양한 콘텐츠에 대하여 안정적이고 정확하게 동영상 재생기의 꼭지점을 검출하는 것을 알 수 있다. 제안하는 기법의 오차는 네 꼭지점을 합하여 평균 7.91 화소로 각 꼭지점에 대해 약 2화소 내외의 오차를 보였다. 반면에 60프레임의 버퍼를 사용하여 시간적 밝기 차이만을 이용한 방법 [3]의 오차 합은 평균 64.6 화소로 한 꼭지점 당 약 16화소 내외의 비교적 큰 오차를 보였다.

IV. 결 론

본 논문에서는 영상 신호의 시공간적 특성을 이용하여 모니터 화면 내의 재생기를 검출하는 방법을 제안하였다. 먼저 시간적 차이 정보와 잡음 제거를 통해 초기 영역을 추정하고, 공간적 특징인 에지 정보와 꼭지점 필터를 이용해 재생기의 네 꼭지점을 정확하게 추출한다. 제안하는 기법은 연산 속도 측면에서 약 0.2초 내외의 빠른 검출 속도로 실시간 동작이 가능하다. 또한 정확도 측면에서는 평균 2화소 내외의 낮은 오차를 가져 기존 기법보다 뛰어난 성능을 보였다.

본 논문에서 제안하는 동영상 재생기 검출 기법은 모니터나 스마트 TV 내에서 실시간 어플리케이션으로 활용될 수 있다.

참 고 문 헌

[1] T. Wiegand, G. Sullivan, G. Bjntegaard and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7 pp. 560-576, Jul.

2003.
 [2] I. Richardson, H.264 and MPEG-4 video compression: video coding for next generation multimedia, Wiley 2003.
 [3] 엘지디스플레이 주식회사, 액정표시장치와 그 구동 방법, 출원번호 10-2007-0059555.
 [4] 삼성전자주식회사, 디스플레이장치에서 하이라이팅 영역을 하이라이팅시킬 수 있는 방법 및 장치, 출원번호 10-2002-0015905.
 [5] R. Gonzalez and R. Woods, Digital image processing, Prentice-Hall, 2002.
 [6] C. Harris and M. Stephens, "A combined corner and edge detector," Proc. Alvey Vision Conference, pp. 147-151, 1988.
 [7] D. Lowe, "Distinctive image features from scale invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

저 자 소 개



김 학 구(학생회원)
 2012년 인하대학교 전자공학과 학사 졸업.
 2012년~현재 인하대학교 전자공학과 석사 과정.
 <주관심분야 : 영상처리, 선명도 개선, 3차원 영상처리>



송 병 철(평생회원)
 1994년 한국과학기술원 전기 및 전자공학과 졸업 (학사).
 1996년 한국과학기술원 전기 및 전자공학과 졸업 (석사).
 2001년 한국과학기술원 전기 및 전자공학과 졸업 (공학박사).

2001년~2008년 삼성전자 디지털미디어연구소 책임연구원

2008년~현재 인하대학교 전자공학부 교수
 <주관심분야: 영상 신호처리, 영상시스템/SoC>